WHAT IS CLAIMED IS:

1. A computer-implemented method for loading objects in a heterogeneous multiprocessor computer system, said method comprising:

identifying a processor to execute a software task, the identification based upon characteristics of the software task and computing resource availability;

loading software code corresponding to the identified processor into a shared memory, wherein the shared memory is shared by a plurality of dislike processors that includes the identified processor; and

executing the loaded code by the identified processor.

2. The method as described in claim 1 further comprising:

prior to the identifying, compiling a source program into at least two object files, each adapted to be executed on a different processor selected from the plurality of dislike processors, wherein the software code that is loaded and executed is one of the object files.

3. The method as described in claim 2 further comprising:

analyzing the source program for program characteristics; and

storing the program characteristics.

4. The method as described in claim 3 wherein at least one of the program characteristics is selected from

the group consisting of data locality, computational intensity, and data parallelism.

5. The method as described in claim 3 wherein identifying the processor further comprises:

retrieving the program characteristics;

retrieving current system characteristics, wherein the current system characteristics includes processor load characteristics for the plurality of dislike processors; and

combining the program characteristics and the current system characteristics to determine which of the dislike processors to assign the software task.

6. The method as described in claim 5 wherein at least one of the current system characteristics is selected from the group consisting of processor availability for each of the dislike processors, and a data size of data being processed by the software task.

7. The method as described in claim 1 further comprising:

determining that the identified processor has a scheduler for scheduling tasks for the processor; and

scheduling the software code to execute on the identified processor, the scheduling including:

writing a software code identifier corresponding to the software code to a run queue corresponding to the identified processor.

8. The method as described in claim 1 further comprising:

signaling the identified processor;

reading, by the identified processor, the software code from the shared memory into a local memory corresponding to the identified processor; and

executing the software code by the identified processor.

9. The method as described in claim 8 further comprising:

writing an instruction block in the shared memory, the instruction block including the address of the loaded software code and the address of an input buffer; and

reading the software code and the input buffer from the locations identified in the instruction block to the identified processor 's local memory.

10. The method as described in claim 9 further comprising:

signaling the identified processor from one of the other processors, the signaling including:

> writing the address of the instruction block to a mailbox that corresponds to the identified processor; and

reading, by the identified processor, the instruction block in response to the signal.

11. An information handling system comprising:

a plurality of heterogeneous processors;

a common memory shared by the plurality of heterogeneous processors;

a first processor selected from the plurality of processors that sends a request to a second processor, the second processor also being selected from the plurality of processors;

a local memory corresponding to the second processor;

a DMA controller associated with the second processor, the DMA controller adapted to transfer data between the common memory and the second processor's local memory; and

a loading tool for loading software code to execute on one of the processors, the loading tool including software effective to:

> identify one of the processors to execute a software task, the identification based upon characteristics of the software task and computing resource availability;
>
> loading the software code corresponding to the identified processor into the common memory; and
>
> executing the loaded code by the identified processor.

12. The information handling system as described in claim 11 further comprising software effective to:

prior to the identification of one of the processors, a source program compiled into at least two object files, each adapted to be executed on a different processor selected from the plurality of heterogeneous processors, wherein the software code that is loaded and executed is one of the object files.

13. The information handling system as described in claim 12 further comprising software effective to:

analyze the source program for program characteristics; and

store the program characteristics.

14. The information handling system as described in claim 13 wherein at least one of the program characteristics is selected from the group consisting of data locality, computational intensity, and data parallelism.

15. The information handling system as described in claim 13 wherein identification of the processor further comprises software effective to:

retrieve the program characteristics;

retrieve current system characteristics, wherein the current system characteristics includes processor load characteristics for the plurality of heterogeneous processors; and

combine the program characteristics and the current system characteristics to determine which of the heterogeneous processors to assign the software task.

16. The information handling system as described in claim 15 wherein at least one of the current system characteristics is selected from the group consisting of processor availability for each of the

heterogeneous processors, and a data size of data being processed by the software task.

17. The information handling system as described in claim 11 further comprising software effective to:

determine that the identified processor has a scheduler for scheduling tasks for the processor; and

schedule the software code to execute on the identified processor, the schedule including software effective to:

write a software code identifier corresponding to the software code to a run queue corresponding to the identified processor.

18. The information handling system as described in claim 11 further comprising software effective to:

signal the identified processor;

read, by the identified processor, the software code from the common memory into a local memory corresponding to the identified processor; and

execute the software code by the identified processor.

19. The information handling system as described in claim 18 further comprising software effective to:

write an instruction block in the common memory, the instruction block including the address of the loaded software code and the address of an input buffer; and read the software code and the input buffer from the locations identified in the instruction block to the identified processor 's local memory.

20. The information handling system as described in claim 19 further comprising software effective to:

signal the identified processor from one of the other processors, the signal including software effective to:

write the address of the instruction block to a mailbox that corresponds to the identified processor; and

read, by the identified processor, the instruction block in response to the signal.

21. A computer program product stored on a computer operable media for loading objects in a heterogeneous multiprocessor computer system, said computer program product comprising:

means for identifying a processor to execute a software task, the identification based upon characteristics of the software task and computing resource availability;

means for loading software code corresponding to the identified processor into a shared memory, wherein the shared memory is shared by a plurality of dislike processors that includes the identified processor; and

means for executing the loaded code by the identified processor.

22. The computer program product as described in claim 21 further comprising:

prior to the means for identifying, means for compiling a source program into at least two object files, each adapted to be executed on a different processor selected from the plurality of dislike processors, wherein the software code that is loaded and executed is one of the object files.

23. The computer program product as described in claim 22 further comprising:

means for analyzing the source program for program characteristics; and

means for storing the program characteristics.

24. The computer program product as described in claim 23 wherein at least one of the program characteristics is selected from the group consisting of data locality, computational intensity, and data parallelism.

25. The computer program product as described in claim 23 wherein the means for identifying the processor further comprises:

means for retrieving the program characteristics;

means for retrieving current system characteristics, wherein the current system characteristics includes processor load characteristics for the plurality of dislike processors; and means for combining the program characteristics and the current system characteristics to determine which of the dislike processors to assign the software task.

26. The computer program product as described in claim 25 wherein at least one of the current system characteristics is selected from the group consisting of processor availability for each of the dislike processors, and a data size of data being processed by the software task.

27. The computer program product as described in claim 21 further comprising:

means for determining that the identified processor has a scheduler for scheduling tasks for the processor; and

means for scheduling the software code to execute on the identified processor, the means for scheduling including:

means for writing a software code identifier corresponding to the software code to a run queue corresponding to the identified processor.

28. The computer program product as described in claim 21 further comprising:

means for signaling the identified processor;

means for reading, by the identified processor, the software code from the shared memory into a local memory corresponding to the identified processor; and means for executing the software code by the identified processor.

29. The computer program product as described in claim 28 further comprising:

means for writing an instruction block in the shared memory, the instruction block including the address of the loaded software code and the address of an input buffer; and

means for reading the software code and the input buffer from the locations identified in the instruction block to the identified processor 's local memory.

30. The computer program product as described in claim 29 further comprising:

means for signaling the identified processor from one of the other processors, the means for signaling including:

means for writing the address of the instruction block to a mailbox that corresponds to the identified processor; and

means for reading, by the identified processor, the instruction block in response to the signal.